



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/448,374	11/23/1999	STEVEN DARDINSKI	102314-46	4969
21125	7590	12/21/2004	EXAMINER	
NUTTER MCCLENNEN & FISH LLP WORLD TRADE CENTER WEST 155 SEAPORT BOULEVARD BOSTON, MA 02210-2604			INGBERG, TODD D	
			ART UNIT	PAPER NUMBER
			2124	

DATE MAILED: 12/21/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application N .	Applicant(s)
	09/448,374	DARDINSKI ET AL.
	Examiner	Art Unit
	Todd Ingberg	2124

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on June 28, 2004.
- 2a) This action is **FINAL**. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-41, 43-76 and 78 is/are pending in the application.
- 4a) Of the above claim(s) 42, 77 and 79-98 is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-41, 43-76 and 78 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) *	4) <input checked="" type="checkbox"/> Interview Summary (PTO-413) Paper No(s)/Mail Date. _____
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)
3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.	6) <input type="checkbox"/> Other: _____.

DETAILED ACTION

Claims 1 - 41, 43-76 and 78 have been examined.

Claims 1, 8, 9, 11, 39, 43, 44, 46, 48, 55, 58, 65, 67, 73, 74 and 78 have been amended

Claims 42, 77, 79-98 have been canceled.

Information Disclosure Statement (IDS)

1. The multiple IDS make up papers 5, 6, 7, 9 and 10. Many non patent literature documents were not considered because Applicant failed to comply with 1.98 Content of Information Disclosure Statement section (b) number (5).

"(5) Each publication listed in an information disclosure statement must be identified by publisher, author (if any), title, relevant pages of the publication, date and place of publication."

Also, all articles must be in English or have a translation.

Applicant should submitted a complete PTO-1449 for just those articles not considered without resubmitting the articles themselves.

Priority

2. Applicant's claim for domestic priority under 35 U.S.C. 119(e) is acknowledged.

Case 60/134,597 has been reviewed. Foreign priority of P11-01503 with a date of 1/21/1999 is the effective filing date.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless -

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 19-37 are rejected under 35 U.S.C. 102(b) as being anticipated by **ControlShell**

User's Manual version 5.1 from Real-Time Innovations, Inc.

published June 1996. (Referred to as **CS** for ControlShell). ControlShell uses the object oriented programming language C++.

Original grounds of rejection ControlShell version 5.0 maintained. Re-mapping of claim limitations required, because of claim amendments

Claim 19

ControlShell version 5.0 anticipates an apparatus for configuring a control system(**CS**, page 1-1, Introduction covers the use of the tool), the apparatus comprising: a plurality of objects (**CS**, page 4-2, shows an inheritance diagram with objects such as "YourComponent", etc) , each of which represents an entity **CS**, page 4-2, a component is an entity) each object being associated with one or more parameters (**CS**, page 4-2, shows an inheritance diagram with objects) , each parameter pertaining to a characteristic of an entity represented by the object (**CS**, page 3-35 to 3-36, the actual screens for building components), at least one object ("descendant" object) being defined as a descendant of another object ("ancestor" object) (**CS**, page 4-2, shows an inheritance diagram with objects) and as being associated with one or more parameters of the ancestor object (Interpreted two ways - Inheritance of the attributes and methods OR the inheritance lines in the figure), a change during configuration to a parameter of an ancestor object being effective as to a descendant object with which that parameter is associated (Interpreted two ways - Inheritance of the attributes and methods OR the inheritance lines in the figure), at least one object being associated with one or more parameter groups, each of which

defines a grouping for one or- more parameters associated with that object (CS, page 4-2, shows an inheritance diagram with objects, CS, grouping in categories and module page 7-32 to 7-34).

Interpretation - The Applicant seems to be using the term "parameter" to mean to different things. On one hand the applicant seems to be referring to the attributes of an object and in another sense the inheritance link can be the interpretation of the term. Assuming the Applicant is being consistent the inheritance of methods and attributes from parent to child class in the inheritance model meets the limitations.

Claim 20

Apparatus according to claim 19, wherein a parameter group defines a grouping with which one or more parameters are presented for any of editing and reporting (CS, pages 5-10 to 5-14).

Claim 21

Apparatus according to claim 20, wherein, as a consequence of descendancy, a descendant object is associated with the parameter groups of the ancestor object from which it descends. As per claim 19.

Claim 22

ControlShell version 5.0 anticipates a method for configuring a control system, the method comprising the steps of: representing entities with objects, each object being associated with one or more parameters, each parameter pertaining to a characteristic of the entity represented by the object, defining at least one object ("descendant" object) as a descendant of another object ("ancestor" object), associating a descendant object with one or more parameters of the ancestor object from which that descendant object descends, and making effective as to that descendant object a change, during configuration, to a parameter of that ancestor object. As per claim 19.

Claim 23

A method according to claim 22, including the step of defining, during configuration, an association between a parameter and an object. As per claim 19.

Claim 24

A method according to claim 23, including the step of defining, during configuration, an object as a descendant of another object. As per claim 19.

Claim 25

A method according to claim 23, wherein each parameter has one or more attributes, and wherein the method includes the step of defining, during configuration (CS, pages 3-34 to 3-36 and the principle of inheritance on instantiation) and the principle of inheritance on instantiation), a value of an attribute . (CS, page 5-26, constructor).

Claim 26

A method according to claim 22, wherein an object represents an entity within any of (i) a controlled system(CS, page 7-3, note components (entities) in figure 7.1), (ii) the control system(CS, page 7-3, note components (entities) in figure 7.1), (iii) a control level hierarchy(CS, page 7-3, note components (entities) in figure 7.1 and in view of the figure page 4-2 for each component, and. (iv) the apparatus for configuring the control system. (CS, ControlShell - the use of this product as per page 1-1).

Claim 27

A method according to claim 26, wherein an entity includes any of a field, control processor block, loop, compound, historian, object type category, object connection, parameter connection,

display placeholder, graphical display entity, and report (CS, page 1-4, Stethoscope CsdBase connection and Chapter 8).

Claim 28

A method according to claim 22, wherein each parameter has one or more attributes, and wherein the attributes of a parameter define any of the following with respect to the characteristic to which the parameter pertains: name, grouping, display label, data type, behavior, help information, edit type, data value range, data value, formula definition, and display format. (CS, pages 5-10 to 5-14 Help information being comments).

Claim 29

A method according to claim 22, including the step of making effective as to a descendant object a change, during configuration, to a parameter of the ancestor object from which the descendant object descends, regardless of whether that change is made before or after the descendant is any of defined and created. (CS, page 4-2, shows an inheritance diagram with objects - Interpreted to be the principle of inheritance and the link made with page 3-36 Base Class Name link. Also, could read on an object outside the inheritance structure with all attributes/methods designated as PUBLIC access).

Claim 30

ControlShell version 5.0 anticipates a method for configuring a control system, the method comprising the steps of: representing entities with objects, each object being associated with one or more parameters, each parameter pertaining to a characteristic of an entity represented by the object, defining at least one object ("descendant" object) as a descendant of another object ("ancestor" object), associating a descendant object with the parameters of the ancestor object

from which that descendant object descends, and making effective as to that descendant object a change, during configuration, to a parameter of that ancestor object. As per claim 19.

Claim 31

A method according to claim 30, comprising the step of associating -a descendant object with parameters in addition to those of the ancestor object from which it descends. As per claim 19.

Claim 32

ControlShell version 5.0 anticipates a method for configuring a control system, the method comprising the steps of representing entities with a plurality of objects, associating each object with one or more parameters as a consequence of any of a parameter definition, parameter override and parameter modification contained or associated with the object As per claim 19, each parameter pertaining to a characteristic of an entity represented by the object, wherein an entity models an entity within any of (i) a controlled system, (ii) the control system, (iii) a control level hierarchy, and (iv) the apparatus for configuring the control system (as per claim 26), defining at least one object as a descendant of another object ("ancestor" object), associating a descendant object with one or more parameters with which the ancestor object is associated, and making effective as to that descendant object a change, during configuration, to a parameter of that ancestor object, except as to a parameter with which the descendant object is associated as a consequence of any of a parameter definition, parameter override and parameter modifications per claim 19 Overriding is part of the principle of polymorphism in object oriented technology and inheritance influences which methods are called by default and which need to be explicitly identified by class).

Claim 33

A method according to claim 32, comprising the steps of defining a second object as a descendant of a first object, and defining a third object as a descendant of -the second object. As per claim 19.

Claim 34

A method according to claim 33, comprising the step of associating the second and third objects with one or more parameters of the first object (as per claim 19), except as to a parameter associated with the second object as a consequence of any of a parameter definition, parameter override and parameter modification. Inheritance as per claim 19.

Claim 35

A method accord to claim 33, associating the third object with a parameter associated with the second object as a consequence of any of a parameter definition, parameter override and parameter modification, regardless of whether a corresponding parameter is associated with the first object. As per claim 19.

Claim 37

A method according to any of claims 30 and 33, wherein an entity includes any of a field device, control processor, block, loop, compound, historian, object type category, object connection, parameter connection, display placeholder, graphical display entity, and report. (CS, page 1-4, Stethoscope CsdBase connection and Chapter 8).

Claim 36

A method according to claim 32, comprising the steps of defining at least one object ("modified" object) as being associated with another object ("modifier" object) for purposes of parameter modification, associating a modified object with one or more parameters of the associated

Art Unit: 2124

modifier object, and making effective as to that modified object a change, during configuration, to a parameter of that modifier object. Interpreted as normal us of object technology and getter and setter methods where parameters(attributes) are retrieved/changed.

Claim 37

A method according to any of claims 30 and 33, wherein an entity includes any of a field device, control processor, block, loop, compound, historian, object type category, object connection, parameter connection, display placeholder, graphical display entity, and report. As per claim 29.

Claim 38

A method according to claim 37, wherein each parameter has one or more attributes defining any of the following with the respect to the characteristic to which the parameter pertains: name, grouping, display label, data type, behavior, help information, edit type, data value range, data value, formula definition, and display format. (CS, pages 5-10 to 5-14 with Help being interpreted as comments).

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless -

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1 - 41, 43-76 and 78 are rejected under 35 U.S.C. 102(b) as being anticipated by

ControlShell User's Manual version 6.0 from Real-Time Innovations, Inc. released March 16, 1998. (Referred to as **CS6** for ControlShell). ControlShell uses the object oriented programming language C++. With the 1998 press release announcing version 6.0 these considered prior art

Art Unit: 2124

under ***In re Epstein*** 31 USPQ2d 1817 (decided August 17, 1994), despite the 1998 copyright date. The Press Release is copied below.

DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

10070016 SUPPLIER NUMBER: 20396899 (THIS IS THE FULL TEXT)
RTI Announces Major New Component-Based Programming System for Building
Complex Electromechanical Systems
PR Newswire, p316SFM064
March 16, 1998
LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 1193 LINE COUNT: 00109

TEXT:

Already Chosen for Critical Application: Shuttle Launch Control
System At Kennedy Space Center

SUNNYVALE, Calif., March 16 /PRNewswire/ -- A component-based, real-time programming system that gives teams building complex electromechanical (EM) systems far more power, capability and time-savings than has been previously available was introduced today by Real-Time Innovations, Inc. (RTI).

Called ControlShell(R), Release 6.0, the system allows the creation of clean, object-oriented designs, built from tested, maintainable, high-quality software components. It combines event-driven logic and sampled-data feedback control in a fully automated visual programming system that greatly reduces design time and improved quality.

After a careful evaluation process, NASA's John F. Kennedy Space Center chose ControlShell 6.0 as the base software architecture for the \$200 million project to redesign the Shuttle Checkout and Launch Control System (CLCS). (See <http://clcs.ksc.nasa.gov>).

Ben Bryant, CLCS Application Software Division Chief at KSC said, "CLCS is using ControlShell to develop our Real Time Control Applications. ControlShell's support of an object-oriented design paradigm fits with the CLCS Application Team's concept of how to maximize reuse and reduce maintenance. ControlShell provides a graphical method to capture data acquisition (Data Flow) and operational control (Finite State Machine) that increases productivity and will allow us to meet our ambitious schedule. ControlShell meets the requirements of our System Engineers (the users) by allowing them to understand the application, through graphical representation, without having to know the underlying language (C++) and provides visibility into the state of their application in the real time operational environment. RTI has provided outstanding support which is

crucial to an important project of this magnitude which I believe has proven beneficial to us both."

According to Dr. Stan Schneider, CEO of Real-Time Innovations, "ControlShell 6.0 offers a crisp, thorough and elegant solution to the challenges that confront designers of complex EM systems. It's the first solution that balances top-down object oriented design and bottom-up component-based synthesis in a true programming system. It also offers a proven framework, excellent visual communications tools, and automated programming. As far as we can tell, nobody provides a comparable solution.

"We are proud to be working closely with KSC on the project to redesign the Shuttle launch control system. It's exciting to launch our product with the hundreds of programmers and engineers who will launch the Shuttle into the next Century."

ControlShell 6.0 allows the creation of EM design from the "top-down" by visually decomposing problems into interacting objects. It also allows designers to build their system from the "bottom-up" by graphically combining components from a repository of reusable, tested and maintainable software.

ControlShell is specially designed for building complex EM systems. It combines event-driven logic, sampled-data feedback control and signal processing. While it allows engineers to build from pre-existing components without coding, it is still an open and powerful programming system that supports the custom software development that truly complex EM projects require.

In addition, its fully-automated visual programming is ideal for team development, providing specific graphical tools that leverage the skills of all the members: system architects, programmers and end users.

ControlShell is supported on Sun, SGI and Windows NT hosts. Real-Time target applications will run on Wind River Systems' VxWorks(R) and SGI's React. In addition, ControlShell is integrated with Wind River Systems Tornado(TM) development environment.

"RTI is one of Wind River's first and closest partners, and provides one of the most popular third party Tornado tools," according to Curt Schacker, Wind River Systems' vice president of marketing. "RTI's new ControlShell 6.0 offers a unique and powerful toolset for complex electromechanical system development. We are pleased to have RTI's newest products integrated with the award winning Tornado development environment."

CONTROLSHELL 6.0 CONCEPTUAL OVERVIEW

Top-Down/Bottom-Up Approach

Traditionally, solutions took one or the other approach to build complex systems. ControlShell does both with equal facility. Top-down design consists of breaking a problem into smaller and smaller parts until the pieces can be easily implemented. It is intuitive, and allows division

of labor and rapid prototyping.

Bottom-up design is the process of synthesizing from proven, pre-existing components. However, it offers no global view to communicate the goals and progress of the project.

Recognizing this trade-off, ControlShell takes a unique merged approach. It allows the user to start with a global, undefined concept, then decompose it into more specialized subsystems. But it goes beyond that to allow easy implementation of subsystems from reusable components.

Structure for Complex Electromechanical Systems

Structural complexity dominates most EM systems, especially when the user begins to connect subsystems with various functions. Such interconnections often lead to even greater complexity, since many systems change behavior in response to changes in operating conditions.

To address this issue, ControlShell provides specific tools and run-time engines that support interface capture and reuse, sampled-data feedback control and event-driven sequencing. ControlShell is a programming system, which means it provides users a framework in which to build their own application. It offers the flexibility that inventive engineers and programmers need to solve difficult and unique problems. In addition, this proven framework allows customers to focus on coding their application and not worry about all the "glue" code, therefore allowing them to get their job done faster.

Complete Automated Programming System

ControlShell recognizes that development teams contain a mix of skills and talents with each team member required to understand the overall problem and contribute specific solutions. ControlShell is a rich toolset that aids programmers and system engineers alike.

The graphical language provides both efficient design and team communication with everything placed in reusable software repositories. Team members then use these repositories to share their work. Installation, code generation, system building and documentation are all automated.

PRICE AND AVAILABILITY

Development licenses for ControlShell start at \$13,250. It is available now.

Real-Time Innovations, Inc. (RTI), a 1991 spin-off from Stanford University, is a leading developer of new tools and architectures for the growing real-time software market. The company's products help real-time developers analyze and understand embedded systems, speed development of distributed real-time systems, and coordinate the work of teams of programmers developing large projects. The company's products are currently in use in thousands of companies worldwide.

For more information, contact Real-Time Innovations, Inc. at 155A Moffett Park Drive, Suite 111, Sunnyvale, CA 94089. Phone: 408-720-8312, FAX: 408-734-5009. World Wide Web site at <http://www.rti.com>.

Art Unit: 2124

SOURCE Real-Time Innovations, Inc.

-0- 03/16/98

/CONTACT: Jenny Simons, 408-720-8312, ext. 205, or jenny@rti.com, or Tom Mahon, 510-937-4921, or tmahon@rti.com, both of Real-Time Innovations, Inc./

/Web site: http://www.rti.com and http://clcs.ksc.nasa.gov/

CO: Real-Time Innovations, Inc. ST: California IN: CPR SU: PDT

CM-KF -- SFM064 -- 8547 03/16/98 18:00 EST http://www.prnewswire.com

COPYRIGHT 1998 PR Newswire Association, Inc.

COMPANY NAMES: Real-Time Innovations Inc.--Product introduction

INDUSTRY CODES/NAMES: BUS Business, General; BUSN Any type of business

DESCRIPTORS: Computer software industry--Product introduction

PRODUCT/INDUSTRY NAMES: 7372513 (Application Development Software)

SIC CODES: 7372 Prepackaged software

FILE SEGMENT: NW File 649

Claim 1

ControlShell version 6.0 anticipates an apparatus for configuring a control system (**CS6**, page 1-1, Introduction covers the use of the tool), the apparatus comprising: a plurality of objects (**CS6**, pages 4-1 to 4-2, object model , each of which represents an entity **CS6**, page 4-2, a component is an entity), each object being associated with one or more parameters (**CS6**,pages 4-2 to 4-3 , parameters) , each parameter pertaining to a characteristic of the entity represented by the object (**CS6**,pages 4-2 to 4-3 , object model and parameters), at least one object ("descendant" object) being defined as a descendant of another object ("ancestor" object) (**CS6**, page 4-2 to 4-3 object model and page 11-10 describes the principle of inheritance in object diagrams) and being associated with one or more parameters of the ancestor object (Interpreted two ways - Inheritance of the attributes and methods OR the inheritance lines in the figure), a change during configuration to a parameter of an ancestor object being effective as to a descendant object with which that parameter is associated (**CS6**, page 4-2 to 4-3 object model and page 11-10 describes the principle of inheritance in object diagrams) without recompilation of those objects

(CS6, page D-8 – Dynamic Binding and page 1-4 Run Time Execution in view of the features dynamic binding applies as per APP – Chapter 7, COG – Chapter 8, FSM – Chapter 9 and the Runtime Interface of Chapter 14 as a modifier).

Claim 2

Apparatus according to claim 1, including an editor that facilitates definition, during configuration, of an association between an parameter and an object. (CS6, pages 4-2 to 4-3 , object model and parameters).

Claim 3

Apparatus according to claim 2, including functionality that facilitates definition (CS6, page 3-36, Methods shown are built in execute ... etc), during configuration, of an object as a descendant of another object (CS6, pages 1-10 to 1-11, 3-20 to 3-22, 3-26 to 3-28, 5-34 to 5-41, 5-43, 5-46, 7-7).

Claim 4

Apparatus according to claim 2, wherein each parameter has one or more attributes, and wherein the apparatus has an editor that facilitates definition, during configuration, of a value of an attribute (CS6, pages 4-2 to 4-3 , object model and parameters – parameters are attributes of an object - pages 1-10 to 1-11, 3-20 to 3-22, 3-26 to 3-28, 5-34 to 5-41, 5-43, 5-46, 7-7).

Claim 5

Apparatus according to claim 1, wherein an object represents an entity within any of (i) a controlled system (As per claim 1), (ii) the control system (As per claim 1, APP, FSM and COG), (iii) a control level hierarchy (As per claim 1 inheritance and page 1-11), and (iv) the apparatus for configuring the control system (As per claim 1).

Claim 6

Apparatus according to claim 5, wherein an entity includes any of a field device, control processor, block, loop, compound, historian, object type category, object connection, parameter connection, display placeholder, graphical display entity, and report. (**CS6**, page D-8 – Dynamic Binding and page 1-4 Run Time Execution in view of the features dynamic binding applies as per APP – Chapter 7, COG – Chapter 8, FSM – Chapter 9 and the Runtime Interface of Chapter 14 as a modifier, Stethoscope page 2-11, scope page 7-23, 15-8 to 15-9, connection pages 1-10 to 1-11, 1-13, 3-8 to 3-10, 3-13, 3-15, 3-19 to 3-23, 3-26 to 3-27, diagram view pages 5-2 to 5-3).

Claim 7

Apparatus according to claim 1, wherein each parameter has one or more attributes, and wherein the attributes of a parameter define any of the following with the respect to the characteristic to which the parameter pertains: name, grouping, display label, data type, behavior, help information, edit type, data value range, data value, formula definition, and display format (**CS6**, pages 7-23, 13-24, 14-3, 14-9 to 14-11 and 14-27).

Claim 8

Apparatus according to claim 1, wherein a change during configuration to a parameter of an ancestor object is effective as to a descendant object with which that parameter is associated, regardless of whether that change is made before or after the descendant is any of defined and created. (As per claim 1 and 6) and wherein that change is effective without recompilation of those objects (as per claim 1).

Claim 9

ControlShell version 6.0 anticipates an apparatus for configuring a control system, the apparatus comprising: a plurality of objects, each object being associated with one or more parameters, each parameter pertaining to a characteristic of an entity represented by the object, at least one object ("descendant" object) being defined as a descendant of another object ("ancestor" object) and as being associated with the parameters of the ancestor object, a change during configuration to a parameter of an ancestor object being effective as to a descendant object with which that parameter is associated , without recompilation of whose objects,

A parameter with which an object is associated as a consequence of any of a parameter definition, parameter override and parameter modification takes precedence over a parameter with which an object is associated as a consequence of being defined as a descendant of another object. The limitations of claim 9 are taught in claim 1.

Claim 10

Apparatus according to claim 9, wherein a descendant object is associated with the parameters of the ancestor object from which it descends, and is associated with further parameters as consequence one or more parameters definitions contained in, or associated with, the descendant object. The limitations are taught by the principle of inheritance in claim 1.

Claim 17

Apparatus according to any of claims 9 and 11, wherein an entity includes any of a field device, control processor, block, loop, compound, historian, object type category, object connection, parameter connection, display placeholder, graphical display entity, and report. As taught in claim 5.

Claim 18

Apparatus according to claim 17, wherein each parameter has one or more attributes defining any of the following with the respect to the characteristic to which the parameter pertains: name, grouping, display label, data type, behavior, help information, edit type, data value range, data value, formula definition, and display format. (As per claim 6 and 7)

Claim 11

ControlShell version 6.0 anticipates an apparatus for configuring a control system, the apparatus comprising: a plurality of objects, each object being associated with one or more parameters, each parameter pertaining to a characteristic of an entity represented by the object (as taught in claim 1), wherein an object represents an entity within any of (i) a controlled system, (ii) the control system, (iii) a control level hierarchy, and (iv) the apparatus for configuring the control system (as taught in claim 5), at least one object being associated with a parameter as a consequence of any of a parameter definition, parameter override and parameter modification contained or associated with the object, at least one object being a descendant of another object ("ancestor" object) and being associated with one or more parameters with which the ancestor object is associated such that changes during configuration to a parameter of an ancestor object is effective as to a descendent object with which that parameter is associated without recompilation of those objects (As per claim 1) , a parameter with which an object is associated as a consequence of any of a parameter definition, parameter override and parameter modification takes precedence over a parameter with which an object is associated as a consequence of being defined as a descendant of another object The limitations are taught by the principle of inheritance in claim 1.

Claim 12

Apparatus according to claim 11, wherein a change during configuration to a parameter of an ancestor object is effective as to a descendant object with which that parameter is associated, except insofar as that parameter is associated with the descendant object as a consequence of any of a parameter definition, parameter override and parameter modification. As per claim 1.

Claim 13

Apparatus according to claim 12, comprising a second object that is defined as a descendant of a first object, and a third object defined as a descendant of the second object. As per claim 1.

Claim 14

Apparatus according to claim 13, wherein a parameter associated with the second object as a consequence of any of a parameter definition, parameter override and parameter modification takes precedence as to the second and third objects over a corresponding parameter associated with the first object. As per claim 1.

Claim 15

Apparatus accord to claim 13, wherein a parameter associated with the second object as a consequence of any of a parameter definition, parameter override and parameter modification is associated with the third object as a consequence of descendancy, regardless of whether a corresponding parameter is associated with the first object. As per claim 1.

Claim 16

Apparatus according to claim 11, wherein at least one object ("modified" object) is associated with another object ("modifier" object) for purposes of parameter modification, and wherein the modified object associated with one or more parameters of the modifier object. Interpreted as

Art Unit: 2124

normal us of object technology and getter and setter methods where parameters(attributes) are retrieved/changed. As per claim 1.

Claim 17

Apparatus according to any of claims 9 and 11, wherein an entity includes any of a field device, control processor, block, loop, compound, historian, object type category, object connection, parameter connection, display placeholder, graphical display entity, and report. As taught in claim 5.

Claim 18

Apparatus according to claim 17, wherein each parameter has one or more attributes defining any of the following with the respect to the characteristic to which the parameter pertains: name, grouping, display label, data type, behavior, help information, edit type, data value range, data value, formula definition, and display format. As per claims 6 and 7.

Claim 39

ControlShell version 6.0 anticipates a method for configuring a control system, the method comprising the steps of: representing entities with objects, each object being associated with one or more parameters, each parameter pertaining to a characteristic of an entity represented by the object, defining at least one object ("descendant" object) being defined as a descendant of another object ("ancestor" object), associating a descendant object with one or more parameters of the ancestor object from which that descendant object descends, and making effective as to that descendant object a change, during configuration, to a parameter of that ancestor object, associating at least one object with one or more parameter groups, each of which defines a

grouping for one or more parameters associated with that object without recompilation of those objects. As per claim 1.

Claim 40

A method according to claim 39, comprising the step of presenting one or more parameters of an object during any of editing and reporting as a function of a parameter group associated with that object. As per claim 20.

Claim 41

A method according to claim 39, comprising associating a descendant object with the ancestor object from which that descendant object descends. As per claim 1.

Claim 43

ControlShell version 6.0 anticipates a method for configuring a control system, the method comprising the steps of: representing entities with objects, each object being associated with one or more parameters, each parameter pertaining to a characteristic of the entity represented by the object, defining at least one object ("descendant" object) as a descendant of another object ("ancestor" object), associating a descendant object with one or more parameters of the ancestor object from which that descendant object descends, and making effective as to that descendant object a change, during configuration, to a parameter of that ancestor object, configuring the control system in accord with one or more of the objects without recompilation of those objects.

As per claim 1.

Claim 44

ControlShell version 6.0 anticipates a apparatus for configuring a process control system, the apparatus comprising: a plurality of objects, each of which represents an entity selected from the

group of entities including a block, block definition, modifier block, modifier block definition, block collection, composite block definition, I/O block, loop template, simple loop, and template derived loop (CS6, Chapter 8), each object being associated with one or more parameters, each parameter pertaining to a characteristic of the entity represented by the object, at least one object ("descendant" object) being defined as a descendant of another object ("ancestor" object) and being associated with one or more parameters of the ancestor object, a change during configuration to a parameter of an ancestor object being effective as to a descendant object with which that parameter is associated without recompilation of those objects. As per claim 1.

Claim 45

Apparatus according to claim 44, wherein each parameter has one or more attributes, and wherein the attributes of a parameter define any of the following with the respect to the characteristic to which the parameter pertains: name, grouping, display label, data type, behavior, help information, edit type, data value range, data value, formula definition, and display format.

As per claim s 6 and 7.

Claim 46

ControlShell version 6.0 anticipates an apparatus for configuring a process control system, the apparatus comprising: a plurality of objects (as per claim 1), each of which represents an entity selected from the group of entities including (see Chapter 9 for the components in a FSM) a block, block definition, modifier block, modifier block definition, block collection, composite block definition, I/O block, loop template, simple loop, and template derived loop, each object being associated with one or more parameters (as per claims 6 and 7), each parameter pertaining to a characteristic of an entity represented by the object (as per claim 1), at least one object

("descendant" object) being defined as a descendant of another object ("ancestor" object) and as being associated with the parameters of the ancestor object (as per claim 1), a change during configuration to a parameter of an ancestor object being effective as to a descendant object with which that parameter is associated without recompilation of those objects. (as per claim 1).

Claim 47

Apparatus according to claim 46, wherein a descendant object is associated with the parameters of the ancestor object from which it descends, and is associated with further parameters as consequence one or more parameters definitions contained in, or associated with, the descendant object. As per claim 1.

Claim 48

ControlShell version 6.0 anticipates an apparatus for configuring a process control system, the apparatus comprising: a plurality of objects (as per claim 1), each of which represents an entity selected from the group of entities (see Chapter 9 for components of FSM) including a block, block definition, modifier block, modifier block definition, block collection, composite block definition, I/O block, loop template, simple loop, and template derived loop, each object being associated with one or more parameters, each parameter pertaining to a characteristic of an entity represented by the object (as per claim 1), wherein an object represents an entity within any of (i) the controlled process, (ii) the control system, (iii) a control level hierarchy, and (iv) the apparatus for configuring the control system (as per claim 5), at least one object being associated with a parameter as a consequence of any of a parameter definition, parameter override and parameter modification contained or associated with the object (as per claim 10), at

least one object being a descendant of another object ("ancestor" object) and being associated with one or more parameters with which the ancestor object is associated (as per claim 1), Such that change during configuration to a parameter of an ancestor object is effective as to a descendant object with which that parameter is associated without recompilation of those objects, (as per claim 1)a parameter with which an object is associated as a consequence of any of a parameter definition, parameter override and parameter modification takes precedence over a parameter with which an object is associated as a consequence of being defined as a descendant of another object (as per claim 1).

Claim 49

Apparatus according to claim 48, wherein a change during configuration to a parameter of an ancestor object is effective as to a descendant object with which that parameter is associated (as per claim 1), except insofar as that parameter is associated with the descendant object as a consequence of any of a parameter definition, parameter override and parameter modification (as per claim 1).

Claim 50

Apparatus according to claim 49, comprising a second object that is defined as a descendant of a first object, and a third object defined as a descendant of the second object (as per claim 1).

Claim 51

Apparatus according to claim 50, wherein a parameter associated with the second object as a consequence of any of a parameter definition, parameter override and parameter modification takes precedence as to the second and third objects over a corresponding parameter associated with the first object. As per claim 1.

Claim 52

Apparatus accord to claim 50, wherein a parameter associated with the second object as a consequence of any of a parameter definition, parameter override and parameter modification is associated with the third object as a consequence of decendancy, regardless of whether a corresponding parameter is associated with the first object. As per claim 1.

Claim 52

Apparatus according to claim 48, wherein at least one object ("modified" object) is associated with another object ("modifier" object) for purposes of parameter modification, and wherein the modified object associated with one or more parameters of the modifier object. As per claim 1.

Claim 54

Apparatus according to claim 48, wherein each parameter has one or more attributes defining any of the following with the respect to the characteristic to which the parameter pertains: name, grouping, display label, data type, behavior, help information, edit type, data value range, data value, formula definition, and display format. As per claims 6 and 7.

Claim 55

ControlShell version 6.0 anticipates an apparatus for configuring a process control system, the apparatus comprising: a plurality of objects, each of which represents an entity selected from the group of entities including a block, block definition, modifier block, modifier block definition, block collection, composite block definition, I/O block, loop template, simple loop, and template derived loop, each object being associated with one or more parameters, each parameter pertaining to a characteristic of an entity represented by the object, at least one object ("descendant" object) being defined as a descendant of another object ("ancestor" object) and as

being associated with one or more parameters of the ancestor object, a change during configuration to a parameter of an ancestor object being effective as to a descendant object with which that parameter is associated, at least one object being associated with one or more parameter groups, each of which defines a grouping for one or more parameters associated with that object without recompilation of those objects. As per claim 46.

Claim 56

Apparatus according to claim 55, wherein a parameter group defines a grouping with which one or more parameters are presented for any of editing and reporting. As per claim 20.

Claim 57

Apparatus according to claim 56, wherein, as a consequence of descendancy, a descendant object is associated with the parameter groups of the ancestor object from which it descends. As per claim 19.

Claim 58

ControlShell version 6.0 anticipates a method for configuring a process control system, the method comprising the steps of: representing entities with objects, each of which represents an entity selected from the group of entities including a block, block definition, modifier block, modifier block definition, block collection, composite block definition, I/O block, loop template, simple loop, and template-derived loop (As per claims 6 and 7) , each object being associated with one or more parameters, each parameter pertaining to a characteristic of the entity represented by the object, defining at least one object ("descendant" object) as a descendant of another object ("ancestor" object), associating a descendant object with one or more parameters of the ancestor object from which that descendant object descends, and making effective as to

that descendant object a change, during configuration, to a parameter of that ancestor object
without recompilation of those objects. As per claims 1 and 46.

Claim 59

A method according to claim 58, including the step of defining, during configuration, an association between an parameter and an object. As per claim 1.

Claim 60

A method according to claim 59, including the step of defining, during configuration, an object as a descendant of another object. As per claim 1.

Claim 61

A method according to claim 59, wherein each parameter has one or more attributes, and wherein the method includes the step of defining, during configuration, a value of an attribute. As per claim 4.

Claim 62

A method according to claim 58, wherein an object represents an entity within any of (i) the control system, (ii) a control level hierarchy, and (iii) the apparatus for configuring the control system. As per claim 5.

Claim 63

A method according to claim 58, wherein each parameter has one or more attributes, and wherein the attributes of a parameter define any of the following with the respect to the characteristic to which the parameter pertains: name, grouping, display label, data type, behavior, help information, edit type, data value range, data value, formula definition, and display format.

As per claims 6 and 7.

Claim 64

A method according to claim 58, including the step of making effective as to a descendant object a change, during configuration, to a parameter of the ancestor object from which the descendant object descends, regardless of whether that change is made before or after the descendant is any of defined and created. As per claim 8.

Claim 65

ControlShell version 6.0 anticipates a method for configuring a process control system, the method comprising the steps of: representing entities with objects (as per claim 1), each entity including any of a (see chapter 9 for implementation of FSM) block, block definition, modifier block, modifier block definition, block collection, composite block definition, I/O block, loop template, simple loop, and template-derived loop, each object being associated with one or more parameters, each parameter pertaining to a characteristic of an entity represented by the object, defining at least one object ("descendant" object) as a descendant of another object ("ancestor" object), associating a descendant object with the parameters of the ancestor object from which that descendant object descends, and making effective as to that descendant object a change, during configuration, to a parameter of that ancestor object without recompilation of those objects. (As per claim 1).

Claim 66

A method according to claim 65, comprising the step of associating a descendant object with parameters in addition to those of the ancestor object from which it descends. As per claim 1.

Claim 67

ControlShell version 6.0 anticipates a method for configuring a process control system, the method comprising the steps of: representing entities with objects, the entities including any of a block, block definition, modifier block, modifier block definition, block collection, composite block definition, I/O block, loop template, simple loop, and template-derived loop, associating each object with one or more parameters as a consequence of any of a parameter definition, parameter override and parameter modification contained or associated with the object, each parameter pertaining to a characteristic of an entity represented by the object, wherein an entity models an entity within any of (i) the control system, (ii) a control level hierarchy, and (iii) the apparatus for configuring the control system defining at least one object as a descendant of another object ("ancestor" object), such that change during configuration to a parameter of an ancestor object is effective as to a descendant object configuration with which that parameter is associated without recompilation of those objects (as per claim 1), associating a descendant object with one or more parameters with which the ancestor object is associated, and making effective as to that descendant object a change, during configuration, to a parameter of that ancestor object (as per claim 46), except as to a parameter with which the descendant object is associated as a consequence of any of a parameter definition, parameter override and parameter modification (as per claim 11).

Claim 68

A method according to claim 67, comprising the steps of defining a second object as a descendant of a first object, and defining a third object as a descendant of the second object. As per claim 1.

Claim 69

A method according to claim 68, comprising the step of associating the second and third objects with one or more parameters of the first object, except as to a parameter associated with the second object as a consequence of any of a parameter definition, parameter override and parameter modification. As per claim 1.

Claim 70

A method accord to claim 68, associating the third object with a parameter associated with the second object as a consequence of any of a parameter definition, parameter override and parameter modification, regardless of whether a corresponding parameter is associated with the first object. As per claim 1.

Claim 71

A method according to claim 67, comprising the steps of defining at least one object ("modified" object) as being associated with another object ("modifier" object) for purposes of parameter modification, associating a modified object with one or more parameters of the associated modifier object, and making effective as to that modified object a change, during configuration, to a parameter of that modifier object. As per claim 1.

Claim 72

A method according to claim 71, wherein each parameter has one or more attributes defining any of the following with the respect to the characteristic to which the parameter pertains: name, grouping, display label, data type, behavior, help information, edit type, data value range, data value, formula definition, and display format. As per claim 7.

Claim 73

ControlShell version 6.0 anticipates a method for configuring a process control system, the method comprising the steps of: representing entities with objects, the entities including any of a block, block definition, modifier block, modifier block definition, block collection, composite block definition, I/O block, loop template, simple loop, and template-derived loop, each object being associated with one or more parameters, each parameter pertaining to a characteristic of an entity represented by the object, defining at least one object ("descendant" object) being defined as a descendant of another object ("ancestor" object), associating a descendant object with one or more parameters of the ancestor object from which that descendant object descends, and making effective as to that descendant object a change, during configuration, to a parameter of that ancestor object, associating at least one object with one or more parameter groups, each of which defines a grouping for one or more parameters associated with that object, without recompilation of those objects. As per claim 46.

Claim 75

A method according to any of claims 73 and 74, comprising the step of presenting one or more parameters of an object during any of editing and reporting as a function of a parameter group associated with that object. As per claim 1.

Claim 76

A method according to any of claims 73 and 74, comprising associating a descendant object with the ancestor object from which that descendant object descends. As per claim 1.

Claim 74

ControlShell version 6.0 anticipates a method for configuring a process control system, the method comprising the steps of: representing entities with objects (as per claim 1), the entities including any of a (See Chapter 8 for defining a FSM) block, block definition, modifier block, modifier block definition, block collection, composite block definition, I/O block, loop template, simple loop, and template-derived loop (As per claims 6 and 7) , each object being associated with one or more parameters, each parameter pertaining to a characteristic of an entity represented by the object (as per claim 1), defining at least one object ("descendant" object) being defined as a descendant of another object ("ancestor" object), associating a descendant object with one or more parameters of the ancestor object from which that descendant object descends (as per claim 1), changing, during configuration, a parameter of that ancestor object, the change being effective as to a descendant object with which that parameter is associated without recompilation of those objects(CS6, page D-8 – Dynamic Binding and page 1-4 Run Time Execution in view of the features dynamic binding applies as per APP – Chapter 7, COG – Chapter 8, FSM – Chapter 9)., associating at least one object with one or more parameter groups, each of which defines a grouping for one or more parameters associated. As per claim 1.

Claim 75

A method according to any of claims 73 and 74, comprising the step of presenting one or more parameters of an object during any of editing and reporting as a function of a parameter group associated with that object. As per claim 1.

Claim 76

A method according to any of claims 73 and 74, comprising associating a descendant object with the ancestor object from which that descendant object descends. As per claim 1.

Claim 78

ControlShell version 6.0 anticipates a method for configuring a process control system, the method comprising the steps of: representing entities with objects, the entities including any of a block, block definition, modifier block, modifier block definition, block collection, composite block definition, I/O block, loop template, simple loop, and template-defined loop, each object being associated with one or more parameters, each parameter pertaining to a characteristic of the entity represented by the object, defining at least one object ("descendant" object) as a descendant of another object ("ancestor" object), associating a descendant object with one or more parameters of the ancestor object from which that descendant object descends, and making effective as to that descendant object a change, during configuration, to a parameter of that ancestor object, configuring the process control system in accord with one or more of the objects without recompilation of those objects. As per claim 46.

Response to Arguments

5. Applicant's arguments are directed toward the amended claims and have been considered but are moot in view of the new ground(s) of rejection for the amended claims. The amendment to claims to distinguish the claimed invention from the commercial product **ControlShell** version 5.1 from Real-Time Innovations, Inc., is presumed to be an amendment made based on Applicant's knowledge of prior art, In re Nilssen 7 USPQ2d 1500 (Fed, Cir. 1988). However, ControlShell version 5.1 uses dynamic binding as described on pages 2-2 to 2-3 and explicitly

states “ ... many instances without recompilation.” Version 6 is more concise and employs COGs.

Conclusion

6. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

7. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

A. David A. Taylor reference teaches the use of COG's from 1992

Correspondence Information

8. Any inquiry concerning this communication or earlier communications from the Examiner should be directed to **Todd Ingberg** whose telephone number is **(703) 305-9775**. The Examiner is working a Maxi-Flex schedule and can be reached Monday through Friday. If

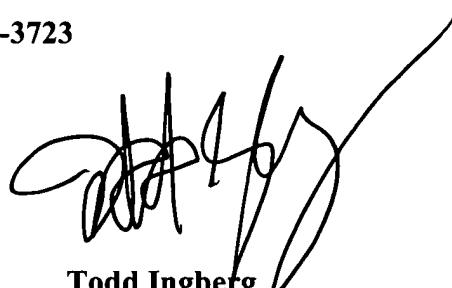
Art Unit: 2124

attempts to reach the examiner by telephone are unsuccessful, the **Examiner's Supervisor**, **Kakali Chaki** be reached at (703)305-9662. The new FAX number is 703 872-9306.

Special Notice

9. Please, Note the Examiner's telephone number will change in October when the Art Unit moves to the new location. The Examiner's new telephone number will be as follows:

(571) 272-3723



Todd Ingberg
Primary Examiner
Art Unit 2124
September 26, 2004